Ross Video Limited

SNMP Support in the openGear Frame

Confidential — For openGear Partners

SNMP Support in the openGear Frame

Revision 01 August 8, 2007 M. Meunier

- Document creation

Restrictions on Use, Duplication, or Disclosure of Proprietary Information

Warranty and Limit of Liability

Copyright

SNMP support in the openGear frame

SNMP manager program used to test our MIBS:
ByteSphere OidView http://www.oidview.com/

SNMP support:

There are 3 major components:

1. **SNMP agent** - this lives on the MFC card and is available through an MFC upgrade file (field upgradeable). With this latest version of software, there is a new tab on the frame controller page to allow you to enable and configure the agent.

   SNMP is a keyed feature that will be sold to customers. To enable SNMP support on a frame, please provide the hardware ID of your frame to Ross Video who will then return the software activation key for this feature. Then, simply enter this software activation key on the frame controller page to enable SNMP.

2. **SNMP support on your cards**. You must support 3 new "get" messages on the CAN bus:

   ```
   // Messages to request SNMP mappings
   #define CANMSG_GET_SNMP_BASE   0x56  // get the SNMP base OID
   #define CANMSG_GET_SNMP_OID    0x57  // get the SNMP OID for a param
   #define CANMSG_GET_SNMP_TRAP   0x58  // get the SNMP TRAP for a param
   ```

   CANMSG_GET_SNMP_BASE:
   You can ignore the content of the request. Response format:
   byte 0:0 return code = 0 (or CANRESP_UNSUPPORTED if not SNMP support)
   byte 1:1 length of the string below, including terminating \0
   byte 2:N SNMP OID of your MIB, relative to the SNMP enterprise node

   The base OID would normally start with your corporate enterprise number, then include some nodes to indicate product type. For example, the base OID for the UDC-8225 is:
   "27399.1.1.3" = rossvideo(27399).openGear(1).openGearObjects(1).udc8225Card(3)
   The initial 6 nodes (to get to your enterprise number) are assumed (1.3.6.1.4.1).

   CANMSG_GET_SNMP_OID:
   Request format is exactly the same as CANMSG_GET_PARAM. Response format:
   byte 0:0 return code = 0 or CANRESP_PARAM_NOTFOUND (if this param is not mapped on SNMP)
   byte 1:2 parameter ID
   byte 3:3 length of the string below, including terminating \0
   byte 4:N SNMP OID for this parameter, relative to your base OID

   The MIB for each openGear card must be structured as one or more tables. Each table is a set of entries, where each entry is a structure holding values for a given card. Therefore the SNMP OID for each parameter (relative to your base OID) is typically a 4-node string.

   The tables are indexed using frame and slot Ids. The agent on the frame controller card handles this automatically. The frame index is for future extension (when multiple frames are accessed via one IP address). With the existing MFC card, all cards will be reported as being in frame 1.

For the UDC, we would report firmware revision as "1.2.1.4" =
udc8225CardObjects(1).udc8225CardTable(2).udc8225CardEntry(1).udc8225FirmwareRev(4)

CANMSG_GET_SNMP_TRAP:
Request format is exactly the same as CANMSG_GET_PARAM.
Response format is the same as for CANMSG_GET_SNMP_OID.

Traps only need to be defined for params where you want the management system to be notified when they change. We have typically defined only 2 or 3 traps per card, to represent input state, reference state, and hardware state. For all other params you should just return CANRESP_PARAM_NOTFOUND. When the agent detects a change in a parameter with an associated trap, it will issue the trap, which will include the SNMP OID and value of the parameter that changed.

Each trap OID is typically a 2-node string, e.g. "0.3" =
udc8225CardNotifications(0).udc8225InputStatusEvent(3)

3. **SNMP MIB definition**. This is a text file, which defines the MIB in ASN.1 syntax. You must define the MIB itself as a subnode of your enterprise (probably a couple branches down as in the example above. Then you must define sub-nodes for notifications(0), objects(1) and conformance(2).

   Underneath notifications, you must define all of your traps, including the bound parameter.

   Underneath objects, you must define
   i) a parameter to hold the number of cards of this type are in the frame (as node 1)
   ii) one or more tables (indexed by frame and slot index) holding the data for each card

   Underneath the table, you must define a card entry (structure holding the data for a card)
   Underneath the entry, you must define each specific parameter. The parameter definitions must be consistent with those from the card.
   CHOICE param maps to enumerated INTEGER, RANGE param maps to Integer32, STRING param maps to DisplayString, etc. The MIBs are organized as multiple tables similar to the DashBoard menu grouping. This makes it easier to find what you want.

   Underneath conformance, you must define groups and compliances, which define what is essential for the agent to implement for this MIB, and any allowable exceptions.

   It is essential that the MIB definition agree closely with the mappings reported by the card, or your SNMP management software will not understand what is being reported by the agent. Not that the agent does not have access to the MIB, so it just has to forward info based on the mapping defined by the card.

   Ross Video MIBs are currently available for reference on the secure website for partners.